Amendments to the Specification

Please amend paragraph 24 as follows:

[0024] According to other aspects of the invention, the encoder memory may store a threshold value. Preferably, the threshold valu[[~~v~~]]e corresponds to a desired minimum compression rate. If such a threshold is provided, the processor determines if a value corresponding to the number of characters in the matching one or more characters is equal to or greater than the threshold value. Only if the threshold is met or exceeded is the second sequence of characters, including the stored predefined compression code, generated.

Please amend paragraph 28 as follows:

[0028] Preferably, the raster image processor stores the predefined compression code, and converts the first sequence of characters by reading a first character in a first sequence of characters, and determining if the read character represents the white or black image data. If so, the raster image processor reads one or more characters occurring immediately subsequent to the first character in the first sequence of characters and determines if the read one or more characters match the read first character. If so, the raster image processor proceeds to generate the second sequence of characters to represent the matching one or more characters.

Please amend paragraph 46 as follows:

[0046] In a second embodiment of the invention, encoding can be interrupted and a more efficient compression technique may be applied. The ~~invention~~ system chooses to interrupt the processing if the stream contains a section of all black or all white data. As a stream of sequential data is processed prior to encoding, if the immediately preceding character, which has yet to be encoded, matches the next character in the stream, and this next character is either solid black (e.g., a stream of all 1s), or solid white (e.g., a stream of all 0s), encoding is interrupted. During the interruption, a determination is made as to whether ~~the invention determines if~~ one or more characters, immediately following the next character in the sequence, also match the next character.

Please amend paragraph 51 as follows:

[0051] Furthermore, the *AGFA* technique provides a look-ahead function. The look-ahead function determines if the sub-string is greater than a minimum number, preferably 6 bytes, and if so, the sub-string is encoded with a new code; the new code comprising any applicable pre-existing code and the length of the code field. The length of the code field is the width of the pre-existing code, with the length forming the most significant bits and serving as a continuation indicator. Any new code follows the length; the new code forming the least significant bits. Like the LZW technique, sub-strings are initially defined by codes having 9 bits, but may be increased to up to 12 bits to add new codes. Once the 12-bit limit is exceeded, the dictionary is reset and subsequent codes are again defined initially with 9 bits.

Please amend paragraphs 53-55 as follows:

[0053] Referring also to FIG. 6, u~~U~~sing the *AGFA* technique, the RIP processor 2050a first sets a reset code 102 (read from the code dictionary 1300) and reads 6000 the "a" in the sequence (which is not black or white 6010) and the "b" immediately thereafter 6020. The RIP processor 2050a then determines 6025 from the code dictionary 1300, if a code exists for the character sequence "ab". Since, in this example, no such code exists at this point in the processing, a new code 105 is generated 6035 to represent the new pattern string "ab" and the new code is stored 6045 in the code dictionary 1300 in memory 2050b. Since the b is not black or white 6050, t~~T~~he RIP processor 2050a continues and reads 6020 the "c" immediately following the "b" in the sequence. The RIP processor 2050a determines 6025 if a code exists for the character sequence "bc". Since, in this example, no such code exist at this point in the processing, a new code 106 is generated 6035 to represent the new pattern string "bc" and the new code is stored 6045 in the dictionary 1300.

[0054] Still a~~A~~lso referring to FIG. 6, since the "c" is not black or white 6050, the RIP 2050 continues and reads 6020~~0~~ the "0" immediately following the "c" in the sequence. The RIP processor 2050a determines if a code exists 2025 for the character sequence "c0". Since, in this example, no such code exists at this point in the processing, a new code 107 is generated 6035 to represent the new pattern string "c0" and the new code is stored 6045 in the dictionary 1300. Also, because the "0" is recognized as a special character 605~~1~~0, the RIP processor 2050a automatically scans ahead 606~~2~~0 to read the next character to determine if it matches 608~~3~~0 the initial "0". If the next character is not a matching "0," the scanning ahead is immediately discontinued and the RIP processor 2050a proceeds with normal processing 602[[4]]0. If the next character 606~~5~~0 is a matching "0," 608~~6~~0 the scanning ahead continues on, character by character 60605~~0~~-60806~~0~~, until no matching "0" is found. At that point 60907~~0~~, the scanning ahead is discontinued and normal processing continues.

[0055] In this exemplary application of the *AGFA* technique, the RIP processor 2050a scans ahead and counts the number of repeated "0" or "1" bytes in the sequence. Preferably, a compression threshold is pre-established and stored in the RIP memory 2050b. For example, the threshold might correspond to a 4 to 1 compression rate. If such a threshold is utilized 6080, and the number of repeated "0" or "1" bytes counted is less than the number required to meet or exceed the threshold, e.g. if the sequence consists of only one or two zeros or ones, then a new code would be established for the sequence in the normal manner 6045~~0~~. Only if the number of repeated "0" or "1" bytes counted meets or exceeds the threshold is the sequence encoded 609~~7~~0 using the applicable pre-defined code 100 or 101.